APL/S PROGRAMMING

SPECIAL FUNCTION KEYS:

[space]	Moves the cursor to the right. Scrolling to the next line is automatic leaving a white square in front.
[back]	Backspace. Erases characters. Scrolls up.
[next]	Moves cursor to next logical line. Enters line.
[shift]	Change upper/lower case. Alternate action.
[erase]	Erases the entire line placing white box at end.
[previous]	Displays the line previous to cursor line. Enters line.
[special]	Opens up an empty line below the cursor line. Enters line.
[master control]	Clears all memory.

BARGRAPHING:

BARHeight	Range= +64 to -64. Larger magnitude no.'s don't hurt.	
BARColor	0- Black	8- Grey
	1- Blue	9- Violet
	2- Green	10- Light Green
	3- Aqua	11- Light Blue
	4- Red	12- Orange
	5- Purple	13- Light Purple
	6- Brown	14- Yellow
	7- Silver	15- White

SYSTEM COMMANDS:

EDIT program-name	Enters edit mode. Allows creation or modificati of a program in RAM.
ENDEdit	Terminates the EDIT mode. Last program statemen
ENDProgram	Marks the end of a particular program. (previou
ERASe name	Removes the named variable or program f/memory.
LIST	Lists the names of variables and programs now in program memory. Also shows bytes free.
LOAD	Loads cassette data into workspace.
SAVE	Stores current workspace on cassette tape.
VERIfy	Verifies the integrety of any tape information regardless of what is in RAM at the time.
ONTR/OFFT	Turns trace mode on and off.

APL/S STATEMENTS

FORMAT	NOTES
variable = value variable(subscript)=value variable=value value value....	Assignment of values to a named variable Variables must start with a LETTER but may include numbers, up to 4 characters. Subscripts may be variables. More than one value forms an array.
BARCode value	Value may be a scalar or an array. There are 7 bars. Value may be a variabl
BARHeight value	Value may be a scalar or an array. Value limit is ±64. May be a variable.
"character-constant"	Print litteral statement. Anything except " marks.
value formula value(subscript) value,value,value	The results of each is printed immediate Subscripts may be formula or variables. , mark allows printing on same line.
IF condition THEN block ENDI	If condition equals 1, (rather than 0) then block is executed.
IF condition THEN block 1 ELSE block 2 ENDI	If condition equals 1, then block 1 is executed, else block 2 is executed. (IF = 0)
KEYBoard	Requests input from keyboard. May be used in program or immediate mode. May be operated on by other functions: eg; +/KEYB or KEYB x KEYB.
program-name	Begins execution of the program with the indicated name. Name may be used as a call to subroutine from another program.
WHILe condition DO block ENDWhile	As long as the condition has the value 1 block is executed. Condition must evaluate to 0 or 1. Block represents a seri of 0 or more statements.

APL/S FUNCTIONS

+ value	Returns the posaitve value.
value + value	Returns the sum of the two values.
+/array plus reduction	Returns the sum of all array values.
- value	Returns the negative value.
value - value	Returns the difference of the two values
-/array	Returns the reduction of the array value
value X value	Returns the product of the two values.
X/array	Returns the product of all array values.
÷ value	Returns the reciprocal of value.
value ÷ value	Returns the 1st devided by the 2nd.
÷ / array	Returns 1st ÷ 2nd ÷ 3rd .... etc.
value * value	1st raised to the 2nd power.
*/array	Repeated exponentiation.
, value	Ravel. Creates an array w/one value.
value , value	Concatenation. Joins two scalars to form an array.
array , array	Concatenation. Joins two arrays to form a new array.
function / array	Reduces all array values to a scalar per the function (+,-,÷,X etc.).
! value	Finds the factorial of the value.
! array	Returns the factorial of all array value
ABS value	Returns the absolute value of "value".
ABS array	Absolute value of all array values.
value AND value	Returns a 1 if both are true (1).
AND/array	Returns a 1 if all array values = 1.
ARRAy value	Creates an array of "value" scalars. Value may be a formula or variable. Scalars are initially = 0.
CEIL value	If value = an integer, the integer is returned. Otherwise the next higher int.
COS value	Returns the cosine of the value in Radia: Value may be a scalar or an array.
value EQ value	Returns a 1 if the values are equal.
EQ/value	Value may be an array of 2 no.'s max.
EXP value	Raises e to the "value" power. value may be a scalar or an array.

APL/S FUNCTIONS (cont)

FLOOr value	Returns the largest integer less than or equal to value. May be a scalar or an array.
value GE value GE/array	Compares the 1st to the 2nd. Returns a 1 if true. Array must be 2 scalars max.
value GT value	Greater than only. Same as GE.
INDEx value	Creates an array of "value" scalars whose value increases by 1.
value LE value LE/array	Less than or = to. Same as GE.
value LT value	Less than only. Same as GT.
LOG value value LOG value LOG/array	Returns natural logarithm of value. Log of 2nd to the 1st base. Log of 2nd to 1st base, Log of 3rd to result base, etc, etc.
value MAX value MAX/array	Returns the larger of the two values. Returns the largest array value.
value MIN value MIN/array	Returns the smaller of the two values. Returns the smallest array value.
value MOD value MOD/array	Returns the remainder of 1st ÷ 2nd. See LOG operation.
value NE value NE/array	Values not equal to each other. (1) Array may be 2 scalars max.
NOT value	Returns the compliment of a scalar or an array whose values = 0 or 1 only.
value OR value OR/array	Returns a 1 if either are true. Returns a 1 if any are true.
RAND value RAND/array	Returns a random integer between 1 and value inclusive. Value must be posative.
SIGN value SIGN/array	Returns the sign value of "value". Same for each scalar in the array.
SINe value SIN/array	Returns the sine in radians of value. Same for all no's in the array.
SIZE array	Returns the no. of elements in the arr
TAN value TAN/array	Returns the tangent in radians of "value Same for each element in the array.

APL/S ERROR CODES

SYNTAX ERRORS:

3. An EXIT was found outside a WHILE block.

5. A program or variable name must follow the ERASE or EDIT command.

6. Additional characters found after an immediate mode statement.

13. BARH or BARC must be followed by a valid formulae.

17. No valid term to the right of a 2 argument + or - function.

23. No valid factor to the right of a 2 argument LOG, ÷,*, x,AND,EQ,GE,GT,LE,LT,MAX,MIN,MOD,NE, or OR function.

24. No valid factor to the right of a concatenation (,) function.

35. No reduction operator (/) following a LOG,÷,+,x,AND,EQ, GE,GT,LE,LT,MAX,MIN,MOD,NE, or OR.

38. No valid factor following the reduction operator (/).

39. No valid factor following a 1 argument +,-,LOG,÷, !,ABS,CEIL,COS,EXP,FLOOR,NOT,RAND,SIGN,SIN, or TAN function.

40. No valid factor following a 1 argument ARRAY,INDEX,SIZE, or ravel (,) function.

41. No valid formulae following a left parenthesis.

42. No right parenthesis, not used for subscripting.

45. No valid formulae following a left parenthesis used for subscripting.

46. No right parenthesis for subscripting.

49. No valid formulae following an = sign in an assignment.

62. No valid formulae following an IF or WHILe statement.

64. Invalid additional characters following an edit mode statement.

82. Missing ENDP statement.

84. Found statement following an ENDP statement.

93. No THEN beginning the next line after an IF statement.

98. No ENDI after a THEN or ELSE statement.

101. No DO beginning the next line after a WHILE statement.

APL/S ERROR CODES

SYNTAX ERRORS: (cont.)

103. No ENDW statement after a DO statement.

128. A left " did not have a matching right ".

135. Unrecognized character.

146. Unrecognized character following the ? for a KEYB statement.

147. Unrecognized character following a numeric constant.

157. A negative sign must be followed by a digit or decimal pt.

158. A decimal pt. not followed by a digit must be followed by one.

168. An E in E-notation must be followed by a neg. sign or a digit.

175. Unrecognized character without a preceeding ? encountered for KEYB input.

CAPACITY (storage overflow) ERRORS:

1. Stack overflow or underflow.

2. Maximum number of constants exceeded (60).

3. Buffer or stack has overflowed. Statement is too complex.

4. Number exceeds permissable range (± 3.618502E±75).

5. E-notation exponent is too large.

6. Program is over 255 bytes long.

142. Out of stack space.

156. Out of data space at a request for program or I/O buffer storage.

206. Out of stack space.

220. Out of data space at a request for array, symbol table, or constant table storage.
Insufficient space for intermediate results may trigger this error statement even if there is still significant storage space.

APL/S ERROR CODES

VALUE (evaluation) ERRORS:

1. Attempt to use variable which has not been assigned a value.

2. Evaluated number is too small or too large.

3. A two-argument function has been applied to two arrays of unequal length.

4. The reduction of an empty-array has no identity value.

5. Subscripted variable is not an array.

6. Subscript is out of range or non-integer.

7. Argument to INDEX or ARRAY is not between 0 & 255 or not an integer.

8. Scalar or one element array is required.

9. Value of 0 or 1 is required.

10. Attempt has been made to edit a variable. Only programs can be edited.

11. Attempt has been made to execute a program that is not executable (program has a syntax error).

12. For value1MODvalue2, value 1 exceeds value 2 by too much. For SIN, COS, or TAN, the argument is too large.

13. Factorial requires a non-negative integer argument less than 128.

14. The value 0 cannot be raised to a negative power.

15. For LOG, one of the arguments is less than or = to 0.

16. A positive non-zero integer argument is required for RAND.

17. For value1*value2, value 1 is less than 0 and value 2 is not an integer.

BARGRAPH SCALING SBR

```
PROG SCAL
SCLR=64÷MAX/ABS BIGN
BIGN=BIGN X SCLR
ENDP
```

Typical Use:

1. Input values to be scaled (eg. A,B,C,D,E,F,G).

2. Form the array BIGNumber using input data:
   BIGN=A,B,C,D,E,F,G

3. Call SCALe program:
   SCALe

4. Assign BIGNumber to graph (BIGN has now been scaled):
   BARH BIGN

- -

CONDITIONAL and AND COMMANDS

1.
```
IF (A GT 0)AND(B GT 0)
THEN block
ELSE block
ENDI
```

Value of parenthetical expression:

```
IF A is greater than 0 ...........1
IF A isn't greater than 0.........0
```

Same for the B side of the expression.

IF left AND right parenthesis = 1, then the "THEN" block is executed.

Otherwise the "ELSE" block is executed (if present).